Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **479,** 865–874 (2018)
Advance Access publication 2018 June 11

doi:10.1093/mnras/sty1531

`FastChem`: A computer program for efficient complex chemical equilibrium calculations in the neutral/ionized gas phase with applications to stellar and planetary atmospheres

Joachim W. Stock,[1]★ Daniel Kitzmann,[2] A. Beate C. Patzer[3] and Erwin Sedlmayr[3]

[1]*Department of Chemistry and Environmental Science, Medgar Evers College – City University of New York, 1650 Bedford Avenue, Brooklyn, NY 11235, USA*
[2]*Center for Space and Habitability, University of Bern, Gesellschaftsstrasse. 6, CH-3012 Bern, Switzerland*
[3]*Zentrum für Astronomie und Astrophysik (ZAA), Technische Universität Berlin (TUB), Hardenbergstr. 36, D-10623 Berlin, Germany*

Accepted 2018 June 7. Received 2018 April 11; in original form 2018 June 7

ABSTRACT

For the calculation of complex neutral/ionized gas-phase chemical equilibria, we present a semi-analytical, versatile, and efficient computer program, called `FastChem`. The applied method is based on the solution of a system of coupled non-linear (and linear) algebraic equations, namely the law of mass action and the element conservation equations including charge balance, in many variables. Specifically, the system of equations is decomposed into a set of coupled nonlinear equations in one variable each, which are solved analytically whenever feasible to reduce computation time. Notably, the electron density is determined by using the method of Nelder and Mead at low temperatures. The program is written in object-oriented C++ which makes it easy to couple the code with other programs, although a stand-alone version is provided. `FastChem` can be used in parallel or sequentially and is available under the GNU General Public License version 3 at https://github.com/exoclime/FastChem together with several sample applications. The code has been successfully validated against previous studies and its convergence behaviour has been tested even for extreme physical parameter ranges down to 100 K and up to 1000 bar. `FastChem` converges stable and robust in even most demanding chemical situations, which posed sometimes extreme challenges for previous algorithms.

Key words: astrochemistry – methods: numerical – planets and satellites: atmospheres – stars: atmospheres.

1 INTRODUCTION

The detailed knowledge of the gas-phase chemical composition is of manifold importance in astrophysics and planetary science. For example, it impacts not only the hydrodynamic and thermodynamic structure of stellar and planetary atmospheres, but also influences the spectral appearance of the object of interest by affecting the related radiative transport coefficients of absorption, and spontaneous and induced emission. Moreover, atomic and molecular species are the elementary building blocks of solid state bodies ranging in size from dust particles to rocky planets.

Chemical equilibrium (CE) situations occur, if the chemical time-scale is much shorter than the dynamical time-scale and if photochemical- and cosmic-ray-induced processes can be neglected. Whether the CE approximation is valid for a particular system has to be checked beforehand. In non-CE situations, the mathematical CE solution is often used to define initial and/or boundary conditions for non-CE models. The CE solution furthermore can be used as a reference state for comparison. CE models have been used, for example, to calculate the chemical composition of the atmospheres of cool stars (e.g. Russell 1934; Vardya 1966; Tsuji 1973; Johnson & Sauval 1982) and in dust-driven winds of AGB (asymptotic giant branch) stars (e.g. Gail, Keller & Sedlmayr 1984; Gail & Sedlmayr 1986, 1987; Dominik et al. 1990; Winters, Dominik & Sedlmayr 1994; Ferrarotti & Gail 2001). More recently, CE models were applied to atmospheres of brown dwarfs (Burrows et al. 2002; Marley et al. 2002; Helling et al. 2008a, b) and extrasolar planets (Madhusudhan, Burrows & Currie 2011; Kataria et al. 2014; Morley et al. 2015).

The CE composition is thermodynamically determined by the minimum of the Gibbs free energy of the system (see Denbigh 1955; Aris 1969, for example). Due to the non-linear dependence of the total Gibbs free energy on the number densities of the chemical species involved in combination with linear constraints (ele-

★ E-mail: joachimstock14@gmail.com (JWS)
daniel.kitzmann@csh.unibe.ch (DK)

ment conservation, including charge balance and the requirement for the number densities to be non-negative), the determination of the CE composition is, in general, non-trivial and often computational highly demanding (see standard textbooks by, e.g. van Zeggeren & Storey 1970; Smith & Missen 1982). Therefore, it is essential to develop rapid, efficient. and versatile computer algorithms for the computation of complex chemical equilibria in general and equilibrium solutions of astrophysical objects in particular.

Different classification schemes for such numerical algorithms were proposed by, e.g. Johansen (1967), van Zeggeren & Storey (1970), and Smith & Missen (1982), each of them emphasizing different aspects, such as, e.g. Gibbs free energy method versus equilibrium constant method, treatment of the element abundance constraints and equilibrium conditions, numerical techniques or considerations of the total number of independent variables in terms of stoichiometric and non-stoichiometric algorithms (Smith & Missen 1982). A convenient way of classification is to distinguish between stoichiometric and non-stoichiometric algorithms first and then by the numerical method (i.e. minimization methods or methods solving sets of non-linear algebraic equations). Stoichiometric algorithms generally use reaction extends as free variables (Naphtali 1959, 1960, 1961; Villars 1959; Cruise 1964; Smith & Missen 1968). Although these methods resulted in applications in, e.g. chemical engineering and combustion chemistry (e.g. Wong, Gottlieb & Lussier 2004), they are rarely, if at all, used in astrophysics or planetary atmospheric science, where nonstoichiometric methods based on optimization techniques or the solution of non-linear algebraic equations are preferably applied.

The most common optimization techniques are the methods known as the RAND method (White, Johnson & Dantzig 1957; White, Johnson & Dantzig 1958) and the method of element potentials (Powell & Sarner 1959), which are, for example, the basis of the computer codes SOLGAS (Eriksson 1971), SOLGASMIX (Eriksson 1975), SOLGASMIX-PV (Besmann 1977), ChemSage (Eriksson & Hack 1990), NASA-CEA (Gordon & McBride 1994; McBride & Gordon 1996), STANJAN (Reynolds 1986), and more recently TECA (Venot et al. 2012) and TEA (Blecic, Harrington & Bowman 2016).

Brinkley (1947) developed a general algorithm to find the CE solution by solving a set of non-linear algebraic equations, that is the law of mass action in combination with the atom balance condition (see also Brinkley 1946; Kandiner & Brinkley 1950) by using the Newton–Raphson method (e.g. Deuflhard 2004)[1] amongst others. A method, known as the (original) NASA method, not to be confused with the algorithm implemented in the NASA-CEA code, was developed by Huff, Gordon & Morrell (1951). Zeleznik & Gordon (1960, 1968) showed the computational equivalence between Brinkley's method, the NASA method, and the RAND method. Therefore, the three methods together are sometimes called the BNR method (Smith & Missen 1982). A similar algorithm to Brinkley's method was applied by Russell (1934) to investigate the CE composition of stellar atmospheres. The main difference to the method of Brinkley is that it is restricted to diatomic molecules only and rests on the hierarchical structure of the element abundances. Finding the CE solution with help of the law of mass action in combination with the element conservation by employing the Newton–Raphson method has been refined by, e.g. Tsuji (1964), Vardya (1966), and Gail et al. (1984) in order to take larger molecules and more species into account. Algorithms of that kind have been implemented in codes such as CONDOR (Lodders & Fegley 1993) and GGChem (Woitke et al. 2018 and references therein).

In this paper, we describe the computationally efficient and fast algorithm of the computer code called `FastChem` that calculates the CE composition of the gas phase for given gas pressure, temperature, and specified element distribution. The algorithm is loosely based on the method described by Gail & Sedlmayr (2014). One major difference to Gail & Sedlmayr (2014) is, for example, our treatment of the electron as a chemical species, for which the charge conservation is solved in `FastChem` with the method of Nelder and Mead (Nelder & Mead 1965) at low temperatures.

The `FastChem` program code is written in object-oriented C++ and is especially designed to be easily coupled with other models. The released source code, however, also includes a stand-alone version. `FastChem` can be run either with double or long double floating-point precision. Long double usually[2] offers a higher precision and allows `FastChem` to compute the gas-phase composition at very low temperatures. While calculations in long double precision normally require a longer computation time per iteration, the total run time until convergence is reached, can still be shorter than that of calculations in double precision. That is because the higher accuracy of long double floating-point precision can result in less iteration steps to be required. Written in an object-oriented way, several instances of `FastChem` can be used simultaneously within one model. For example, a double precision version of `FastChem` can be run for high temperatures, whereas a long double precision instance can be run for low temperature at the same time. This also allows `FastChem` to be used in parallel by employing, e.g OpenMP (Open Multi-Processing) (openmp.org 1997), which greatly increases the computational speed. The chemistry model can also be parallelized with MPI (mcs.anl.gov 1997) by creating a separate `FastChem` instance on each MPI node, for example. The `FastChem` code is released as open-source under the GNU General Public License version 3 (gnu.org 2007). The source code, together with several examples demonstrating various possible applications of `FastChem`, is published at: https://github.com/exoclime/FastChem.

2 METHOD

Let $\mathcal{S}$ be the set of all species in the gas phase without the electron and $\mathcal{E} \subset \mathcal{S}$ be the set of all chemical elements taken into account in the particular model. Furthermore, let $\mathcal{S}_0$ be the set of all species and $\mathcal{E}_0 \subset \mathcal{S}_0$ be the set of all elements with the electron included.

The number densities n_i for all species $S_i \in \mathcal{S}_0$ composed of elements $E_j \in \mathcal{E}_0$ for a given total gas pressure p_g and a given temperature T, are determined in dissociative equilibrium

$$S_i \rightleftharpoons \nu_{i0}E_0 + \nu_{i1}E_1 + \nu_{i2}E_2 + \ldots + \nu_{ij}E_j + \ldots = \sum_{j\in\mathcal{E}_0} \nu_{ij}E_j \tag{1}$$

by the law of mass action

$$n_i = K_i \prod_{j\in\mathcal{E}_0} n_j^{\nu_{ij}} \qquad \forall i \in \mathcal{S} \setminus \mathcal{E} \tag{2}$$

[1]For a historical note, see (Kollerstrom 1992)

[2]Note that the actual accuracy of double vs. long double depends on the employed compiler and computer platform. In some cases (e.g. Visual C++ compiler on a Windows operating system), long double and double will provide the same accuracy.

which can be derived by minimizing the Gibbs free energy of the system (see, e.g. Denbigh 1955; Aris 1969) in combination with the equations for element, respectively, charge conservation

$$\epsilon_j n_{\langle \mathrm{H} \rangle} = n_j + \sum_{i \in \mathcal{S} \setminus \mathcal{E}} \nu_{ij} n_i \qquad \forall j \in \mathcal{E}_0 \, . \tag{3}$$

Here, K_i denotes the mass action constant, ϵ_j the relative elemental abundance with respect to hydrogen, ν_{ij} the coefficients of the stoichiometric matrix, and

$$n_{\langle \mathrm{H} \rangle} = n_{\mathrm{H}} + \sum_{i \in \mathcal{S} \setminus \mathcal{E}} \nu_{i\mathrm{H}} n_i \tag{4}$$

the sum of all hydrogen nuclei per unit volume. The index 0 denotes the electron by definition, i.e. $\epsilon_0 = 0$ owing to the charge neutrality. All stoichiometric coefficients ν_{ij} are nonnegative integers, if $j \neq 0$. For positively charged species, ν_{i0} is a negative integer number; for negatively charged species, ν_{i0} is a positive integer. Otherwise, ν_{i0} is 0. Any meaningful solution of the problem requires that the number densities n_i are nonnegative

$$n_i \geq 0 \tag{5}$$

which poses an additional constraint for the mathematical solution $\left\{n_0, \ldots, n_{|\mathcal{S}_0|}\right\}$. Equations (2) and (3) form a system of coupled nonlinear algebraic equations. In the next two subsections the input and output data are described followed by the outline of the algorithm in Section 2.4.

2.1 Input data

For the computation of the chemical equilibrium solution, the algorithm needs as input a list of chemical elements, molecules, and/or ions, which can be cations and/or anions of atoms and/or molecules, respectively. Furthermore, the chemical element composition is given by

$$\epsilon_j = 10^{x_j - 12.0}, \tag{6}$$

using here the convention of stellar atmospheric theory, i.e. $x_{\mathrm{H}} = 12$. In addition, the natural logarithm of the dimensionless mass action constant of species $S_i \in \mathcal{S} \setminus \mathcal{E}$

$$\ln \bar{K}_i(T) = -\frac{\Delta_{\mathrm{r}} G_i^{\ominus}(T)}{R\,T} \tag{7}$$

for all considered molecules and ions are required, where $\Delta_{\mathrm{r}} G_i^{\ominus}(T)$ is the Gibbs free energy of dissociation (equation 1) and R the universal gas constant. Gibbs free energies of dissociation $\Delta_{\mathrm{r}} G_i^{\ominus}(T)$ can be calculated via

$$\Delta_{\mathrm{r}} G_i^{\ominus}(T) = \Delta_{\mathrm{f}} G_i^{\ominus}(T) - \sum_{j \in \mathcal{E}_0} \nu_{ij} \Delta_{\mathrm{f}} G_j^{\ominus}(T) \qquad i \in \mathcal{S} \setminus \mathcal{E} \tag{8}$$

with the Gibbs free energies of formation $\Delta_{\mathrm{f}} G_i^{\ominus}(T)$ of species S_i, adopted from thermochemical databases such as, e.g. Chase (1998). The mass action constants can be interpolated lookup tables which might prove to be time and memory consuming. Here, we prefer to fit the data with the expression

$$\ln \bar{K}_i(T) = \frac{a_0}{T} + a_1 \ln T + b_0 + b_1 T + b_2 T^2 \tag{9}$$

which we derived by using the van't Hoff equation (van't Hoff 1884; Atkins & de Paula 2014) and Kirchoff's law of thermochemistry (Kirchhoff 1858; Atkins & de Paula 2014) in combination with the quadratic expansion in temperature of the heat capacity $C_p^{\ominus}(T)$. Thermochemical data are fitted within a prescribed temperature range depending on the available data. Applying polynomial fits outside the the temperature range can lead to improper results (Burcat 1984), especially if higher orders are involved. Higher order terms are avoided in equation (9) so that extrapolation is at least possible in a limited range (see discussion by Woitke et al. 2018 and references therein). New chemical species can be easily added, if their mass action constants are available, or removed by simply modifying the list of species in the input file. The program code can also be adapted quite easily to a user-specified input data format.

If requested, the user can provide the relative atomic masses of the elements $A_{\mathrm{r},j}$ for the additional calculation of the mean relative molecular mass according to

$$\left\langle M_{\mathrm{r},i} \right\rangle = \frac{1}{n_{\mathrm{g}}} \sum_{i \in \mathcal{S}} n_i \, M_{\mathrm{r},i} \simeq \frac{1}{n_{\mathrm{g}}} \sum_{j \in \mathcal{E}} \epsilon_j \, n_{\langle \mathrm{H} \rangle} \, A_{\mathrm{r},j} \qquad , \tag{10}$$

where $M_{\mathrm{r},i}$ is the relative molecular mass.

Finally, the user has to specify at least one pair of (p_{g}, T)-data, where p_{g} is the gas pressure. Alternatively and computationally faster, the user can choose to provide the total pressure of hydrogen nuclei

$$p_{\langle \mathrm{H} \rangle} = n_{\langle \mathrm{H} \rangle} \, k_{\mathrm{B}} \, T \tag{11}$$

instead of p_{g}. The Boltzmann constant is denoted by k_{B}.

2.2 Output data

The output data is given in a formatted file listing the number densities n_i of all species $S_i \in \mathcal{S}$. In four separate columns, the gas pressure p_{g}, the temperature T, the total gas density n_{g}, and $n_{\langle \mathrm{H} \rangle}$ are added in the output file for visualization of the results with scientific information graphics software. If requested, the output file also contains the mean relative molecular mass based on the specified relative atomic masses.

Moreover, the `FastChem` program generates a monitor file, listing the total number of pressure iterations, the total number of chemistry iterations in the last pressure iteration step, and information about convergence. We recommend to carefully examine this file after the calculation is completed.

2.3 Gas pressure – total hydrogen nuclei density conversion

If $p_{\langle \mathrm{H} \rangle}$ is provided as input parameter for the calculation, then $n_{\langle \mathrm{H} \rangle}$ is easily obtained by equation (11) However, for most practical applications, a total gas pressure p_{g} provided by the user rather than $p_{\langle \mathrm{H} \rangle}$, is used as input. In this case, the gas pressure is converted into $n_{\langle \mathrm{H} \rangle}$ by the following procedure. In H-He-rich gas mixtures at moderate temperatures $n_{\langle \mathrm{H} \rangle}$ can be approximated by

$$n_{\langle \mathrm{H} \rangle} \approx n_{\mathrm{H}} + 2\, n_{\mathrm{H}_2} \tag{12}$$

which can be used to estimate the number density of He atoms

$$\epsilon_{\mathrm{He}} n_{\langle \mathrm{H} \rangle} \approx n_{\mathrm{He}} \, . \tag{13}$$

Thereby, the contributions of other molecules, bearing less abundant elements and ions are neglected in both equations (equations 12 and 13). The total gas number density is then simply obtained by

$$n_{\mathrm{g}} \approx n_{\mathrm{H}} + n_{\mathrm{H}_2} + n_{\mathrm{He}} \tag{14}$$

which yield a relation between $n_{\langle \mathrm{H} \rangle}$ and the total gas density n_{g}

$$n_{\langle \mathrm{H} \rangle} \approx \frac{1}{2\, b^2 K_{\mathrm{H}_2}} \left(a + 4\, b\, K_{\mathrm{H}_2} n_{\mathrm{g}} - \sqrt{a^2 + 4\, b\, K_{\mathrm{H}_2} n_{\mathrm{g}}} \right) \tag{15}$$

after eliminating $n_{\rm H}$, $n_{\rm H_2}$, $n_{\rm He}$ by using equation (2) for $i = {\rm H_2}$ and introducing the coefficients

$$a = 1 + \epsilon_{\rm He} \tag{16}$$

and

$$b = 1 + 2\,\epsilon_{\rm He}\,. \tag{17}$$

This rough estimate of $n_{<{\rm H}>}$ is used as initial value $n^{(0)}_{\langle {\rm H}\rangle}$ for the following iteration scheme. After the number densities $n_i^{(k)}$, $k > 0$ of all species $S_i \in \mathcal{S}_0$ are determined, the total number density

$$n_{\rm g}^{(k)} = \sum_{i\in\mathcal{S}_0} n_i^{(k)} \tag{18}$$

is used instead of the approximation given by equation (14). If the calculated $n_{\rm g}^{(k)}$ is larger than $p_{\rm g}/(k_{\rm B}\,T)$ provided by the user, then we set

$$n^{(k+1)}_{\langle {\rm H}\rangle} = (1-\lambda)\,n^{(k)}_{\langle {\rm H}\rangle} \tag{19}$$

and

$$n^{(k+1)}_{\langle {\rm H}\rangle} = \frac{1}{(1-\lambda)}\,n^{(k)}_{\langle {\rm H}\rangle} \tag{20}$$

otherwise, where $\lambda \in (0, 1)$ is a damping parameter. To avoid oscillations, we reduce λ, if the expression $\left(p_{\rm g}/(k_{\rm B}\,T) - n_{\rm g}^{(k)}\right)$ changes sign between two iteration steps. The iteration is continued until the convergence criterion

$$\left|\frac{p_{\rm g}}{k_{\rm B}\,T} - n_{\rm g}^{(k)}\right| < \delta\left|\frac{p_{\rm g}}{k_{\rm B}\,T}\right| \qquad \delta > 0 \tag{21}$$

is fulfilled.

2.4 The FastChem algorithm

2.4.1 Preconditioning

The FastChem algorithm follows roughly the idea of the method presented by Gail & Sedlmayr (2014). That is, instead of solving the equations (2) and (3) simultaneously, e.g. with a Newton–Raphson method in higher dimensions, the equation system is decomposed into a set of equations, each of them in one variable n_j, $j \in \mathcal{E}$. Therefore, we rewrite the element conservation equation (3) with help of the law of mass action equation (2) as

$$\epsilon_j n_{\langle {\rm H}\rangle} = n_j + \sum_{k=1}^{N_j} k\,n_j^k \sum_{\substack{i\in\mathcal{S}\setminus\mathcal{E}\\ \nu_{ij}=k\\ \epsilon_i=\epsilon_j}} K_i \prod_{\substack{l\in\mathcal{E}_0\\ l\neq j}} n_l^{\nu_{il}} + n_{j,\rm min} \qquad j\in\mathcal{E} \tag{22}$$

reducing the number of variables from $|\mathcal{S}_0|$ to $|\mathcal{E}_0|$ and solve equation (22) element by element, where

$$n_{j,\rm min} = \sum_{\substack{i\in\mathcal{S}\setminus\mathcal{E}\\ \epsilon_i<\epsilon_j}} \nu_{ij} n_i \qquad j\in\mathcal{E} \tag{23}$$

is the total number density of all species build of elements less abundant than element j,

$$\epsilon_i = \min_{j\in\mathcal{E}}\left\{\epsilon_j \middle| \nu_{ij} \neq 0\right\} \qquad i\in\mathcal{S}\setminus\mathcal{E} \tag{24}$$

and

$$N_j = \max_{i\in\mathcal{S}\setminus\mathcal{E}}\left\{\nu_{ij}\middle|\epsilon_i = \epsilon_j\right\} \qquad j\in\mathcal{E}\,. \tag{25}$$

Solving equation (22) essentially reduces to the problem of finding the root of the polynomial

$$P_j(n_j) := \sum_{k=0}^{N_j} A_{jk} n_j^k \tag{26}$$

of degree N_j for all $j \in \mathcal{E}$ with the coefficients

$$A_{j0} = -\epsilon_j n_{\langle {\rm H}\rangle} + n_{j,\rm min} \tag{27}$$

$$A_{j1} = 1 + \sum_{\substack{i\in\mathcal{S}\setminus\mathcal{E}\\ \nu_{ij}=1\\ \epsilon_i=\epsilon_j}} K_i \prod_{\substack{l\in\mathcal{E}_0\\ l\neq j}} n_l^{\nu_{il}} \tag{28}$$

$$A_{jk} = k \sum_{\substack{i\in\mathcal{S}\setminus\mathcal{E}\\ \nu_{ij}=k\\ \epsilon_i=\epsilon_j}} K_i \prod_{\substack{l\in\mathcal{E}_0\\ l\neq j}} n_l^{\nu_{il}}, \qquad k \geq 2\,. \tag{29}$$

The polynomials $P_j(n_j)$ and if needed their derivatives

$$P_j'(n_j) = \sum_{k=1}^{N_j} k A_{jk} n_j^{k-1} \tag{30}$$

are evaluated by employing Horner's rule (Horner 1819). For the evaluation of the products in equations (28) and (29), see Appendix A.

2.4.2 Computational procedure

To determine the CE composition, FastChem firstly sorts the elements according to their abundance in descending order using an adapted version of the piksr2 algorithm (Press et al. 1992). Afterwards, the set of equations, i.e., equation (22) is solved iteratively, starting with the most abundant element, where $n_{j,\rm min}$ is employed as a correction term. Since $n_{j,\rm min}$ is supposed to be relatively small, we set the initial values $n^{(0)}_{j,\rm min} = 0$ for all elements except of carbon (C) in the carbon-rich case and oxygen (O) in the oxygen-rich case to account for the high bond energy of the CO molecule. In the carbon-rich case, we set $n^{(0)}_{\rm C,min} = \epsilon_{\rm O} n_{\langle {\rm H}\rangle}$, and in the oxygen-rich case, we set $n^{(0)}_{\rm O,min} = \epsilon_{\rm C} n_{\langle {\rm H}\rangle}$. The initial electron density $n_0^{(0)}$ is always set to a very small value. The FastChem algorithm works as follows:

Step 1
Initial values for the electron density $n_0^{(0)}$ and for the correction terms $n^{(0)}_{j,\rm min}$ are set and the logarithmic mass action constants $\ln K_i$ are calculated for a given temperature T.

Step 2
The number densities for all atomic species n_j ($j \in \mathcal{E}$) are calculated via equation (22) (or equation 34 if necessary) in descending order, starting with the most abundant element.

Step 3
The results are used to calculate the number densities of the molecular species n_i ($i \in \mathcal{S} \setminus \mathcal{E}$) via the law of mass action equation (2).

Step 4
$n_{j,\rm min}$ is updated.

Step 5
The electron density n_0 is calculated (see Section 2.4.3).

Steps 2 to 5 are repeated until the convergence criterion

$$\left|n_i^{(k)} - n_i^{(k-1)}\right| < \delta \left|n_i^{(k)}\right| \qquad \delta > 0 \tag{31}$$

is met for all species $i \in \mathcal{S}$.

The procedure depends on whether A_{j0} is a strictly negative number or not. If n_j is the solution of equation (22), it is quite evident that $A_{jN_j} > 0, A_{jk} \geqslant 0$ for $0 < k < N_j$ and $A_{j0} < 0$. However, during the iteration it might happen that A_{j0} becomes positive. This situation can occur, if there are three elements, say X, Y, and Z, with $\epsilon_X \gtrapprox \epsilon_Y$ and $\epsilon_X \gtrapprox \epsilon_Z$ forming two molecules XY and XZ with large mass action constants K_{XY} and K_{XZ}. The molecules XY and XZ, competing for the element X, lead then to a large correction $n_{\mathrm{X,\,min}}$ which can exceed $\epsilon_X n_{<\mathrm{H}>}$.

We first consider the case $A_{j0} < 0$. It can be easily shown that there exists a unique solution of equation (22) which suffice the condition given in equation (5). If N_j is less than three, there are simple analytical expressions which provide the (temporary) solution for the number density n_j, namely

$$n_j = -A_{j0} \begin{cases} 1 & \text{if } N_j = 0 \\ 1/A_{j1} & \text{if } N_j = 1 \\ 2/\left(A_{j1} + \sqrt{A_{j1}^2 - 4A_{j2}A_{j0}}\right) & \text{if } N_j = 2. \end{cases} \tag{32}$$

Otherwise, we employ the classical ordinary Newton–Raphson method (Deuflhard 2004) in one dimension

$$n_j^{(\mu+1)} = n_j^{(\mu)} - \frac{P_j^{(\mu)}}{P_j'^{(\mu)}} \tag{33}$$

to obtain n_j. Since $P_j(n_j)$ is two times continuously differentiable, convex in the interval $[0, \epsilon_j n_{<\mathrm{H}>}]$, $P_j(0) < 0$ and $P_j(\epsilon_j n_{<\mathrm{H}>}) > 0$, there exists only one unique root n_j^* and hence the Newton–Raphson method converges monotonously against the mathematical solution for suitable initial values, i.e. $n_j^{(0)} > n_j^*$. Therefore, no computational costly damping strategy is required. To guarantee convergence of the Newton–Raphson method, we set the initial value $n_j^{(0)} = \epsilon_j n_{\langle\mathrm{H}\rangle}$.

If $A_{j0} \geqslant 0$, then no real root of $P_j(n_j)$ exists, which suffice the condition $n_j > 0$. In this case, equation (22) in Step 2 is replaced by

$$\epsilon_j n_{\langle\mathrm{H}\rangle} = n_j + \sum_{k=1}^{N_j} k\, n_j^k \sum_{\substack{i \in \mathcal{S}\setminus\mathcal{E} \\ \nu_{ij}=k}} K_i \prod_{\substack{l \in \mathcal{E}_0 \\ l \neq j}} n_l^{\nu_{il}} \tag{34}$$

for element $E_j \in \mathcal{E}$ for all remaining FastChem iterations. Moreover $n_{j,\,\mathrm{min}}$ does not need to be updated for that element anymore. If the procedure fails to converge within a prescribed number of iterations, FastChem starts again and tries to find the CE solution by solving equation (34) for all elements $E_j \in \mathcal{E}$. In case this backup procedure fails, the computation is aborted with an error message.

The algorithm computes the CE composition $\{n_0, \ldots, n_{|\mathcal{S}_0|}\}$ for each pair (p_g, T) or $(p_{<\mathrm{H}>}, T)$ separately. For large sets of (p_g, T)-pairs or $(p_{<\mathrm{H}>}, T)$-pairs the computational speed might be increased by using the solutions of neighboring grid points as initial values $n_i^{(0)}$. However, it might happen in such a case, that some of the attributed values/initial values underestimate the number densities n_j, $j \in \mathcal{E}$, of the CE solution, which can impair the convergence behaviour of the Newton solver. Therefore, this approach is not implemented in the FastChem code so far.

Table 1. Solar photospheric element abundances x_j according to Asplund et al. (2009) for all elements included in the test example.

Element		x_j	Element		x_j
H	Hydrogen	12.00	Na	Sodium	6.24
He	Helium	10.93	Ni	Nickel	6.22
O	Oxygen	8.69	Cr	Chromium	5.64
C	Carbon	8.43	Cl	Chlorine	5.50
Ne	Neon	7.93	Mn	Manganese	5.43
N	Nitrogen	7.83	P	Phosphorus	5.41
Mg	Magnesium	7.60	K	Potassium	5.03
Si	Silicon	7.51	Co	Cobalt	4.99
Fe	Iron	7.50	Ti	Titanium	4.95
S	Sulfur	7.12	F	Fluorine	4.56
Al	Aluminium	6.45	Zn	Zinc	4.56
Ar	Argon	6.40	Cu	Copper	4.19
Ca	Calcium	6.34	V	Vanadium	3.93

2.4.3 *Determination of the electron density*

For the electron species, equation (22) becomes a homogeneous equation. Although, at least for systems including only ions of ionization degree one ($|\nu_{i0}| = 1$), an analytic solution can be derived (see Appendix B), we follow a different approach here. If there are sufficient free electrons available, the electron density n_0 can be calculated from the sum of the ion densities

$$n_0 = -\sum_{i \in \mathcal{S}} \nu_{i0} n_i \,. \tag{35}$$

Note that $\nu_{i0} > 0$ for anions and $\nu_{i0} < 0$ for cations. That approach, however becomes problematic at low temperatures and/or high pressures due to cancellation of leading digits. Therefore, if $n_0 < 0.9\, n^+$, we employ the method of Nelder and Mead (Nelder & Mead 1965; Lagrias et al. 1997), also known as the downhill simplex method (Press et al. 1992). The method of Nelder and Mead is designed to find minima of multivariable functions $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ and is especially successful for low dimensions $n \in \mathbb{N}$. Furthermore, no evaluation of the derivative or the Jacobi matrix of f is necessary. We have reduced the method to one dimension. To find the mathematical solution of the electron density n_0, we introduce the objective function

$$f(y_0) := \left| e^{y_0} + \sum_{i \in \mathcal{S}} \nu_{i0} n_i \right| \tag{36}$$

$$= \left| e^{y_0} + \sum_{i \in \mathcal{S}} \nu_{i0} \exp\left\{ \ln K_i + \nu_{i0} y_0 + \sum_{j \in \mathcal{E}} \nu_{ij} \ln n_j \right\} \right| \tag{37}$$

with $y_0 = \ln n_0$. In the second part of the equation, the law of mass action, i.e., equation (2) is used. To dampen numerical oscillations, we modify the the electron density at the iteration step μ according to

$$n_0^{(\mu)} = \sqrt{n_0^{(\mu)} n_0^{(\mu-1)}} \,. \tag{38}$$

3 RESULTS AND DISCUSSION

3.1 Test calculations

In order to validate our code, we compare the results of FastChem with the pure gas-phase results presented by Sharp & Huebner

Table 2. List of all species included in the test model. Mass action constants $\ln \bar{K}_i$ are fitted according to equation (9) using thermochemical data from Chase (1998) unless indicated otherwise.

Element	Molecules, Ions
H	H_2, H^+, H^-, H_2^+, H_2^-
He	He^+
O	HO, HO_2, H_2O, H_2O_2 [a], O_2, O_3, HO^+, HO^-, H_3O^+, O^+, O^-, O_2^+, O_2^-
C	CH, CHO, CH_2, CH_2O, CH_3, CH_4, CO, CO_2, C_2, C_2H, C_2H_2, C_2H_4, C_2H_4O, C_2O, C_3, C_3H [b], C_3O_2, C_4, C_5, C^+, C^-, CH^+, CH^- [b], CHO^+, CO_2^-, C_2^-
Ne	Ne^+
N	CHN, CHNO, CN, CNO, CNN, NCN, C_2N, C_2N_2, C_4N_2, HN, HNO, c-HNO_2, t-HNO_2, HNO_3, H_2N, H_2N_2, H_3N, H_4N_2, NO, NO_2, NO_3, N_2, N_2O, N_2O_3, N_2O_4, N_2O_5, N_3, CN^+, CN^-, N^+, N^-, NO^+, NO_2^-, N_2^+, N_2^-, N_2O^+
Mg	HMg, HMgO, H_2MgO_2, MgN, MgO, Mg_2, $HMgO^+$, Mg^+
Si	CSi, CSi_2, C_2Si [b], C_2Si_2, HSi, H_2Si [b], H_3Si [b], H_4Si, NSi, NSi_2, OSi, O_2Si, Si_2, Si_3, HSi^+, Si^+, Si^-
Fe	C_5FeO_5, FeH_2O_2, FeO, Fe^+, Fe^-
S	COS, CS, CS_2, FeS, HS [c], H_2O_4S, H_2S, MgS, NS, OS, OS_2 [c], O_2S, O_3S, SSi, S_2, S_3, S_4, S_5, S_6, S_7, S_8, HS^- [b], S^+, S^-
Al	AlH, AlHO(1), AlHO(2), $AlHO_2$, AlN, AlO, AlO_2, AlS, Al_2, Al_2O, Al_2O_2, CAl, Al^+, Al^-, $AlHO^+$, $AlHO^-$, AlO^+, AlO^-, AlO_2^-, Al_2O^+, $Al_2O_2^+$
Ar	Ar^+
Ca	CaH [b], CaHO, CaH_2O_2, CaO, CaS, Ca_2, Ca^+, $CaHO^+$
Na	CNNa, $C_2N_2Na_2$, HNa, HNaO, $H_2Na_2O_2$, NaO, Na_2, Na_2O_4S, $HNaO^+$, Na^+, Na^-, NaO^-
Ni	C_4NiO_4, HNi [b], NiO [b], NiS, Ni^+, Ni^-
Cr	C_2Cr [b], CrH [b], CrN, CrO, CrO_2, CrO_3, Cr^+, Cr^-
Cl	AlCl, AlClO, $AlCl_2$, $AlCl_3$, Al_2Cl_6, CCl, CClN, CClO, CCl_2, CCl_2O, CCl_3, CCl_4, CHCl, $CHCl_3$, CH_2Cl_2, CH_3Cl, C_2Cl_2, C_2Cl_4, C_2Cl_6, C_2HCl, CaCl, $CaCl_2$, ClFe, ClH, ClHO, ClH_3Si, ClMg, ClNO, $ClNO_2$, ClNa, ClNi, ClO, ClO_2, ClO_3, ClS, ClS_2, ClSi, Cl_2, Cl_2Fe, Cl_2H_2Si, Cl_2Mg, Cl_2Na_2, Cl_2Ni, ClOCl, ClClO, ClO_2Cl, ClOClO, Cl_2O_2S, Cl_2S, Cl_2Si, Cl_3Fe, Cl_3HSi, Cl_3Si, Cl_4Fe_2, Cl_4Mg_2, Cl_4Si, Cl_6Fe_2, $AlCl^+$, $AlCl_2^+$, $AlCl_2^-$, Cl^+, $ClMg^+$, ClS^+, Cl^-, Cl_2S^+
Mn	HMn [b], MnO [b], MnS [b], Mn^+
P	CHP, CP, ClP, Cl_3OP, Cl_3P, Cl_3PS, Cl_5P, HP, H_2P, H_3P, NP [d], OP [d], O_2P, O_6P_4 [d], $O_{10}P_4$, PS [c], P_2, P_4, P_4S_3, P^+, P^-
K	CKN, $C_2K_2N_2$, ClK, Cl_2K_2, HK, HKO, $H_2K_2O_2$, KO, K_2, K_2O_4S, HKO^+, K^+, K^-, KO^-
Co	ClCo, Cl_2Co, Cl_3Co, Cl_4Co_2, Co^+, Co^-
Ti	C_2Ti [b], C_4Ti [b], ClOTi, ClTi, Cl_2OTi, Cl_2Ti, Cl_3Ti, Cl_4Ti, NTi [b], OTi, O_2Ti, STi [b], Ti^+, Ti^-
F	AlClF, $AlClF_2$, $AlCl_2F$, AlF, AlFO, AlF_2, AlF_2O, AlF_3, AlF_4Na, Al_2F_6, CClFO, $CClF_3$, CCl_2F_2, CCl_3F, CF, CFN, CFO, CF_2, CF_2O, CF_3, CF_4, CF_4O, CF_8S, CHF, CHFO, CHF_3, CH_2ClF, CH_2F_2, CH_3F, C_2F_2, C_2F_3N, C_2F_4, C_2F_6, C_2HF, CaF, CaF_2, ClF, ClFMg, $ClFO_2S$, $ClFO_3$, ClF_2OP, ClF_3, ClF_3Si, ClF_5, ClF_5S, $CHClF_2$, $CHCl_2F$, Cl_2FOP, Cl_3FSi, CoF_2, FFe, FH, FHO, FHO_3S, FH_3Si, FK, FMg, FN, FNO, FNO_2, FNO_3, FNa, FO, FOTi, OFO, FOO, FP, FPS, FS, FSi, FTi, F_2, F_2Fe, F_2H_2, F_2H_2Si, F_2K_2, F_2Mg, F_2N, c-F_2N_2, t-F_2N_2, F_2Na_2, F_2O, F_2OS, F_2OSi, F_2OTi, F_2O_2, F_2O_2S, F_2P, F_2S, $F_2S_2(1)$, $F_2S_2(2)$, F_2Si, F_2Ti, F_3Fe, F_3HSi, F_3H_3, F_3N, F_3NO, F_3OP, F_3P, F_3PS, F_3S, F_3Si, F_3Ti, F_4H_4, F_4Mg_2, F_4N_2, F_4S, F_4Si, F_4Ti, F_5H_5, F_5P, F_5S, F_6H_6, F_6S, F_7H_7, $F_{10}S_2$, $AlClF^+$, AlF^+, AlF_2^+, AlF_2^-, AlF_2O^-, AlF_4^-, CF^+, CF_2^+, CF_3^+, F^+, F^-, FMg^+, FP^+, FP^-, FS^+, FS^-, F_2K^-, F_2Mg^+, F_2Na^-, F_2P^+, F_2P^-, F_2S^+, F_2S^-, F_3S^+, F_3S^-, F_4S^+, F_4S^-, F_5S^+, F_5S^-, F_6S^-
Zn	Zn^+, Zn^-
Cu	ClCu, Cl_3Cu_3, CuF, CuF_2, CuH [b], CuO, CuS [b], Cu_2, Cu^+, Cu^-
V	C_2V [b], C_4V [b], NV, OV, O_2V, V^+, V^-

[a] For H_2O_2, the data by Chase (1998) are only tabulated up to $T = 2000$ K. Extrapolation of $\ln \bar{K}_{H_2O_2}$ to higher temperatures yields good agreement with data obtained by Goos, Burcat & Ruscic (2016).
[b] Thermochemical data by Tsuji (1973) used to fit $\ln \bar{K}_i$.
[c] Thermochemical data from Barin (1995) used to fit $\ln \bar{K}_i$.
[d] Thermochemical data by Burcat & Ruscic (2005) and Goos et al. (2016) used to fit $\ln \bar{K}_i$.

(1990), who, in turn, compared their results with the ones presented by Tarafdar (1987). In our test calculation, we adopt a fixed total hydrogen nuclei pressure $p_{\langle H \rangle} = 1000\,\mathrm{dyn\,cm^{-2}}$ and the solar photospheric element abundances determined by Asplund et al. (2009) for elements more abundant than Vanadium (V) (see Table 1).

The set of species used in our test calculation encompasses all molecules and ions listed by Chase (1998) formed from the elements shown in Table 1. The list of species is complemented by molecules of potential astrophysical interest (Tsuji 1973). Therefore, in contrast to Sharp & Huebner (1990), we use a slightly different set of species, which does not include the low abundant elements Scandium (Sc), Zirconium (Zr), Strontium (Sr), Bromine (Br) and Yttrium (Y). However, our standard scenario includes the noble gases Helium (He), Neon (Ne), and Argon (Ar), the elements Cobalt (Co) and Zinc (Zn), ions, and numerous additional molecules, mostly chlorides and fluorides (see Table 2).

Logarithmic mass action constants $\ln \bar{K}_i$ were determined using thermochemical data from Barin (1995), Burcat & Ruscic (2005), Chase (1998), Goos et al. (2016) and Tsuji (1973) as indicated in Table 2. Barin (1995) and Chase (1998) provide Gibbs free energy data $\Delta_f G_i^{\ominus}(T)$ in tabulated form. From these data $\ln \bar{K}_i$ were calculated according to equations (7) and (8). The obtained mass action constants $\ln \bar{K}_i$ were then subsequently fitted according to equation (9). Burcat & Ruscic (2005), Goos et al. (2016), and Tsuji (1973) provide thermochemical data in form of polynomials in temperature T or reciprocal temperature $\theta = 5040/T$, from which we obtained $\ln \bar{K}_i$ values, which we then fitted to derive the coefficients used in equation (9).

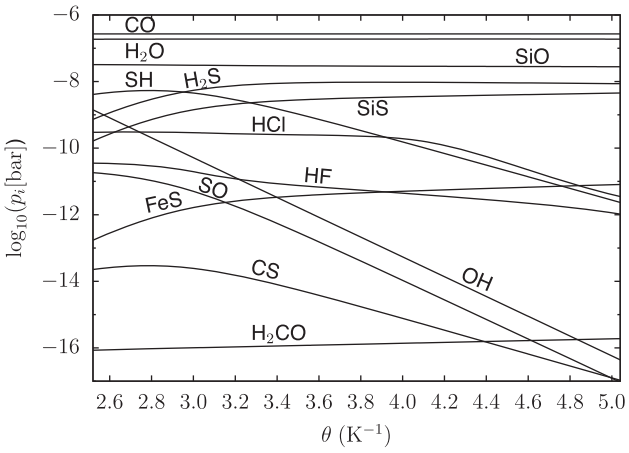

Figure 1. Logarithmic partial pressures of the molecular species CO, H_2O, SiO, SH, H_2S, OH, SiS, HCl, HF, SO, FeS, CS, and H_2CO calculated with `FastChem`. The total hydrogen nuclei pressure $p_{<H>}$ equals 0.01 bar as in the calculations by Tarafdar (1987) and Sharp & Huebner (1990).

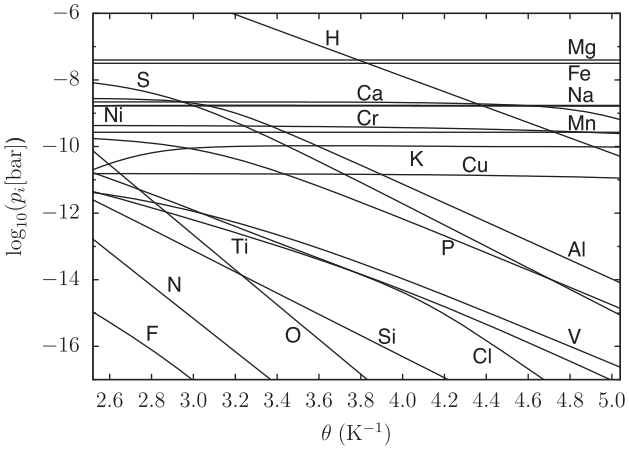

Figure 2. Logarithmic partial pressures of the the atomic species H, N, O, F, Na, Mg, Al, Si, P, S, Cl, K, Ca, Sc, Ti, V, Cr, Mn, Fe, Ni, Cu, Br, Sr. and Zr. The total hydrogen nuclei pressure $p_{<H>}$ equals 0.01 bar as in the calculations by Tarafdar (1987) and Sharp & Huebner (1990).

Fig. 1 shows the same species as presented by Sharp & Huebner (1990) and Tarafdar (1987), respectively.

Overall, we find good qualitative agreement with Tarafdar (1987) and very good qualitative agreement with Sharp & Huebner (1990). Like Sharp & Huebner (1990), we find that in contrast to the results of Tarafdar (1987) H_2S to be more abundant than SiS and FeS more abundant than CS at $\theta = 2.65263\,\mathrm{K}^{-1}$ ($T = 1900\,\mathrm{K}$).

Fig. 2 shows the partial pressures of all atomic species in our model.

Besides the expected deviations due to the updated element abundances ϵ_j used here (cf. Table 1), we find a decrease in the partial pressure of potassium (K) with decreasing reciprocal temperature ($\theta < 3\,\mathrm{K}^{-1}$). The total number density of positively charged ions

$$n^+ = \sum_{\substack{i \in \mathcal{S}\setminus\mathcal{E} \\ \nu_{i0}<0}} n_i \tag{39}$$

as well as the electron density n_0 increase with increasing temperature (Fig. 3, dotted line).

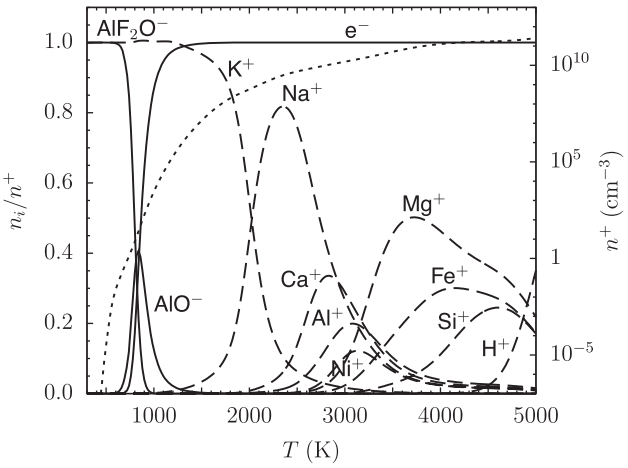

Figure 3. Electron number density and number densities of ions relative to the total number density of cations (dotted line). Electron and anions are marked with solid lines, cations with dashed lines.

Above $T \approx 800\,\mathrm{K}$ the charge balance is determined by free electrons and monatomic cations. The maxima of the positively charged ions are distributed according to the ionization energies of the corresponding elements with the exception of Al and Ca. The exception is due to the temperature dependence of the partition functions. Below $T \approx 800\,\mathrm{K}$ free electrons become very rare and the charge balance is primarily determined by K^+ and the polyatomic anions AlO^- and AlF_2O^- in our set of species shown in (Table 2). Note, that the n^+ already dropped to $1\,\mathrm{cm}^{-3}$. The low ionization energy of potassium (4.3406633 eV, (Sugar & Corliss 1985)) leads to its ionization at relatively low temperatures and a decrease in the number density of potassium with decreasing reciprocal temperature θ (see Fig. 2).

3.2 Convergence behavior

In this section, we study the efficiency of `FastChem` by simply determining how many iterations `FastChem` requires to meet the convergence criterion equation (31) with $\delta = 10^{-6}$ for selected test scenarios. For this numerical test we use the same set of species as in the previous section (see Table 2) but vary the gas pressures p_g between 10^{-12} bar and 10^3 bar and the reciprocal temperature between $2\,\mathrm{K}^{-1}$ and $50\,\mathrm{K}^{-1}$ yielding a grid of 49 600 points. Note, that these extreme parameter ranges are selected exclusively for numerical tests. Typical (p_g, θ) profiles of some astrophysical objects are shown in Fig. 4.

These profiles are for illustration only. An in-depth interpretation with respect to these objects is not recommended since some of these atmospheres are clearly not in chemical equilibrium.

Therefore, `FastChem` was called for each of the 49 600 grid points separately. For a typical (p_g, θ) combination such as $p_g = 10^{-5}$ bar and $\theta = 15\,\mathrm{K}^{-1}$, 6 iterations in the oxygen-rich case are needed to fulfill the convergence criterion. Usual computation times for `FastChem` are in the low milliseconds range for 510 species, including ions, on a standard desktop computer. For smaller systems restricted to neutral and ionized species composed of C, H, O, N, e^- execution times can be reduced to microseconds. Without electrons, the computational time can be considerably smaller. The computational effort can increase to seconds for very low temperatures combined with high pressures.

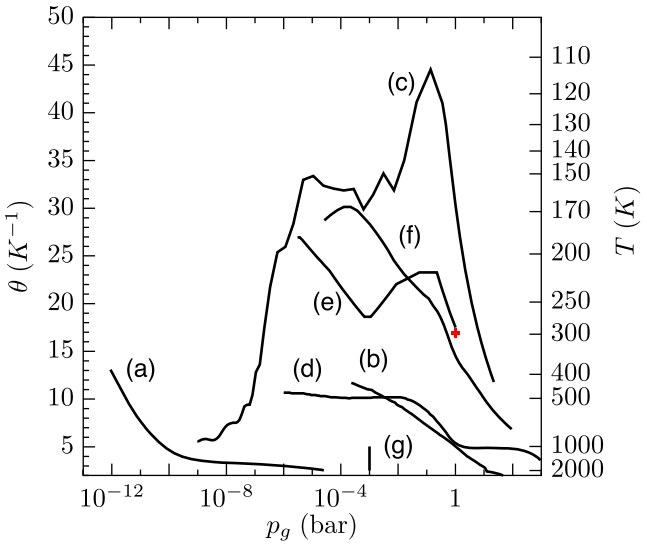

Figure 4. Selected thermal structures. The red cross marks the reference state for thermochemical data. (a) Circumstellar shell of an AGB star (Gail & Sedlmayr 1987), (b) Brown dwarf star GJ229b (Tsuji, Ohnaka & Aoki 1999) (c) Jupiter's atmosphere from measurements by the Galileo probe (Seiff et al. 1998) (d) super-Earth GJ1214b with solar abundances (Miller-Ricci & Fortney 2010), (e) Earth's atmosphere up to 86 km altitude (COESA 1976) (f) Venus reference atmosphere (Seiff et al. 1985) (g) (p_g, θ) data used by Sharp & Huebner (1990) and in Figs 1 and 2.

Three different chemical scenarios are studied in the specified (p_g, θ)-plane:

(i) oxygen-rich (element abundances according to Table 1)
(ii) carbon abundance equals oxygen abundance (i.e. C: O = 1, otherwise Table 1)
(iii) carbon-rich (Table 1, but with the exchanged values of x_C and x_O)

Fig. 5 shows the total number of iterations of the chemistry at the last pressure iteration step as function of p_g and θ.

Convergence is reached for all points in the (p_g, θ) plane shown. In general the number of iterations increases with increasing reciprocal temperature and total gas pressure. However, this behavior is not monotonously. There are some features, where the total number of iterations becomes relatively large, most likely due to the competition between molecules for one element such as e.g. O. The effect is especially pronounced in the cases where C:O $\lessapprox 1$.

While the calculations presented in this work are only focused on hydrogen-rich cases, `FastChem` has been also successfully tested for environments, where hydrogen is only a minor species. However, the current version of `FastChem` requires a comparatively long calculation time for these cases because the pressure iteration is done via $n_{<H>}$ (see Section 2.3). In a future version of `FastChem` we will adapt the code to remove the explicit dependence on hydrogen and replace it with the major element present.

4 SUMMARY

For the efficient calculation of complex gas phase chemical equilibria, we present a semi-analytical, flexible computer program, called `FastChem`. The program is written in object-oriented C++ which makes it easy to couple the code with other programs, although a stand-alone version is provided. `FastChem` can be used in parallel or sequentially and is available under the GNU General

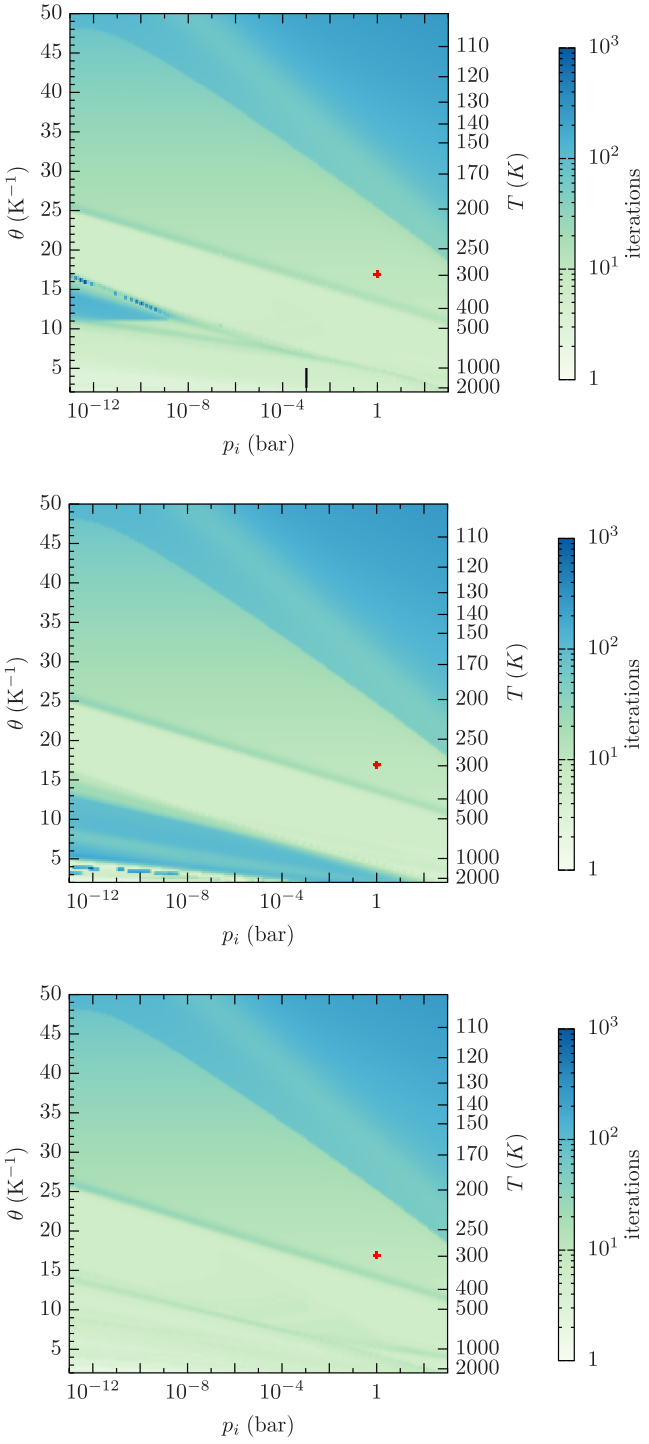

Figure 5. Total number of iterations of the chemistry at the last pressure iteration step. The C:O ratio is $(C:O)_{solar}$ (upper panel), 1 (mid panel) and $1/(C:O)_{solar}$ (lower panel). The red cross marks the reference state for thermochemical data. (p_g, θ) data used in Figs 1 and 2 are marked by a black solid line in the upper panel.

Public License version 3 at https://github.com/exoclime/FastChem together with several sample applications. The code has been successfully validated against previous studies and its convergence behavior has been tested even for extreme physical parameter ranges down to 100 K and up to 1000 bar. `FastChem` shows a stable and robust convergence behavior in even most demanding chemi-

cal situations (e.g. electrons competing with multiple anions at low temperatures and densities, equal carbon and oxygen element abundances, very large molecules consisting of more than 300 atoms), which posed sometimes to be extremely challenging for previous CE codes.

ACKNOWLEDGEMENTS

DK acknowledges financial and administrative support by the Center for Space and Habitability and the PlanetS National Centre of Competence in Research (NCCR). BP acknowledges support by the Deutsche Forschungsgemeinschaft (DFG) Schwerpunktprogramm SPP 1573 - grant PA 1648/2-1.

REFERENCES

Aris R., 1969, Elementary Chemical Reactor Analysis, Prentice-Hall Inc., Englewood Cliffs, NJ
Asplund M., Grevesse N., Sauval A. J., Scott P., 2009, ARA&A, 47, 481
Atkins P., de Paula J., 2014, Physical Chemistry, 10th edition, Oxford University Press, Oxford, UK
Barin I., 1995, Thermochemical Data of Pure Substances, 3rd edition, VCH Verlagsgesellschaft mbH, Weinheim
Besmann T. M., 1977, Technical report, SOLGASMIX-PV, A Computer Program to Calculate Equilibrium Relationships in Complex Chemical Systems. Oak Ridge National Laboratory, Oak Ridge, TN 37830
Blecic J., Harrington J., Bowman M. O., 2016, ApJS, 225, 4
Brinkley S. R., 1946, J. Chem. Phys., 14, 563
Brinkley S. R., 1947, J. Chem. Phys., 15, 107
Burcat A., 1984, Thermochemical Data for Combustion Calculations. Springer US, New York, NY, p. 455–473
Burcat A., Ruscic B., 2005, Third Millennium Ideal Gas and Condensed Phase Thermochemical Database for Combustion with updates from Active Thermochemical Tables, Technion-IIT, Aerospace Engineering, and Argonne National Laboratory, Chemistry Division, Haifa and Lemont IL
Burrows A., Burgasser A. J., Kirkpatrick J. D., Liebert J., Milsom J. A., Sudarsky D., Hubeny I., 2002, ApJ, 573, 394
COESA, 1976, Technical report, U.S. Standard Atmosphere 1976. NOAA, NASA, US Air Force
Chase M., 1998, NIST-JANAF Thermochemical Tables, J. Phys. Chem. Ref. data, Monograph 9, Gaithersburg, MD
Cruise D. R., 1964, J. Phys. Chem., 68, 3797
Denbigh K., 1955, The Principles of Chemical Equilibrium, Cambridge University Press, Cambridge, UK
Deuflhard P., 2004, Newton Methods for Nonlinear Problems – Affine Invariance and Adaptive Algorithms, Springer, Berlin
Dominik C., Gail H. P., Sedlmayr E., Winters J. M., 1990, A&A, 240, 365
Eriksson G., 1971, Acta Chem. Scand., 25, 2651
Eriksson G., 1975, Chem. Scripta, 8, 100
Eriksson G., Hack K., 1990, Metall. Trans. B, 21, 1013
Ferrarotti A. S., Gail H.-P., 2001, A&A, 371, 133
Gail H.-P., Sedlmayr E., 1986, A&A, 166, 225
Gail H. P., Sedlmayr E., 1987, A&A, 171, 197
Gail H.-P., Sedlmayr E., 2014, Physics and Chemistry of Circumstellar Dust Shells, Cambridge University Press, Cambridge UK
Gail H.-P., Keller R., Sedlmayr E., 1984, A&A, 133, 320
Goos E., Burcat A., Ruscic B., 2016, Extended Third Millenium Ideal Gas Thermochemical Database with updates from Active Thermochemical Tables, Technion IIT, Aerospace Engineering, and Argonne National Laboratory, Chemistry Division, Haifa and Lemont, IL
Gordon S., McBride B. J., 1994, Technical report, Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications I. Analysis
Helling C. et al., 2008a, MNRAS, 391, 1854
Helling C., Dehn M., Woitke P., Hauschildt P. H., 2008b, ApJ, 675, L105
Horner W. G., 1819, Phil. Trans. R. Soc. Lond., 109, 308
Huff V. N., Gordon S., Morrell V. E., 1951, Technical report, General Method and Thermodynamic Tables for Computation of Equilibrium Composition and Temperature of Chemical Reactions
Johansen E. S., 1967, Acta Chem. Scand., 21, 2273
Johnson H. R., Sauval A. J., 1982, A&AS, 49, 77
Kandiner H. J., Brinkley S. R., 1950, Ind. Eng. Chem., 42, 850
Kataria T., Showman A. P., Fortney J. J., Marley M. S., Freedman R. S., 2014, ApJ, 785, 92
Kirchhoff G., 1858, Ann. Phys., 179, 177
Kollerstrom N., 1992, British J. Hist. Sci., 25, 347
Lagrias J. C., Reeds J. A., Wright M. H., Wright P. E., 1997, Technical report, Convergence Properties of the Nelder-Mead Simplex Algorithm in Low Dimensions. Computing Sciences Research Center, Bell Laboratories, Murray Hill, NJ 07974
Lodders K., Fegley B., 1993, Earth Planet. Sci. Lett., 117, 125
Madhusudhan N., Burrows A., Currie T., 2011, ApJ, 737, 34
Marley M. S., Seager S., Saumon D., Lodders K., Ackerman A. S., Freedman R. S., Fan X., 2002, ApJ, 568, 335
McBride B. J., Gordon S., 1996, Technical report, Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications II. Users Manual and Program Description. Glenn Research Center – NASA
Miller-Ricci E., Fortney J. J., 2010, ApJ, 716, L74
Morley C. V., Fortney J. J., Marley M. S., Zahnle K., Line M., Kempton E., Lewis N., Cahoy K., 2015, ApJ, 815, 110
Naphtali L. M., 1959, J. Chem. Phys., 31, 263
Naphtali L. M., 1960, in Bahn G. S., Zukoski E. E. Eds., Kinetics, Equilibria and Performance of High Temperature Systems, Proceedings of First Conference, Butterworths, London
Naphtali L. M., 1961, Ind. Eng. Chem., 53, 387
Nelder J., Mead R., 1965, Comput. J., 7, 308
Powell H. N., Sarner S. F., 1959, Technical report, The use of element potentials in the analysis of chemical equilibrium, Vol. 1. General Electric Co.
Press W., Teukolsky S., Vetterling W., Flannery B., 1992, Numerical Recipies in FORTRAN: The Art of Scientific Computing
Reynolds W. C., 1986, Technical Report, The Element-Potential Method for Chemical Equilibrium Analysis: Implementation in the Interactive Program STANJAN. Department of Mechanical Engineering, Stanford University, Stanford, CA 94305
Russell H. N., 1934, ApJ, 79, 317
Seiff A., Schofield J. T., Kliore A. J., Taylor F. W., Limaye S. S., 1985, Adv. Space Res., 5, 3
Seiff A. et al., 1998, J. Geophys. Res., 103, 22857
Sharp C. M., Huebner W. F., 1990, ApJS, 72, 417
Smith W. R., Missen R. W., 1968, Can. J. Chem. Eng., 46, 269
Smith W. R., Missen R. W., 1982, Chemical Reaction Equilibrium Analysis: Theory and Algorithms, Wiley, New York, NY
Sugar J., Corliss C., 1985, J. Phys. Chem. Ref. Data 14, 14, Suppl. 2
Tarafdar S. P., 1987, in Vardya M. S., Tarafdar S. P., eds, IAU Symposium Vol. 120, Astrochemistry, p. 559
Tsuji T., 1964, Ann. Tokyo Astron. Obs., 9, 1
Tsuji T., 1973, A&A, 23, 411
Tsuji T., Ohnaka K., Aoki W., 1999, ApJ, 520, L119
Vardya M. S., 1966, MNRAS, 134, 347
Venot O., Hébrard E., Agúndez M., Dobrijevic M., Selsis F., Hersant F., Iro N., Bounaceur R., 2012, A&A, 546, A43
Villars D. S., 1959, J. Phys. Chem., 63, 521
White W. B., Johnson S. M., Dantzig G. B., 1957, Technical Report, Chemical Equilibrium in Complex MIxtures. RAND Corp., Santa Monica, CA
White W. B., Johnson S. M., Dantzig G. B., 1958, J. Chem. Phys., 28, 751
Winters J. M., Dominik C., Sedlmayr E., 1994, A&A, 288, 255
Woitke P., Helling C., Hunter G. H., Millard J. D., Turner G. E., Worters M., Blecic J., Stock J. W., 2018, A&A, 614, A1
Wong F. C., Gottlieb J. J., Lussier L.-S., 2004, Technical Report, Chemical Equilibrium Mixture Computations for Energetic Material Combustion

in Closed Vessels. Defense Technical Information Center, 8725 John J. Kingman Road, Fort Belvoir, VA 22060-6218
Zeleznik F. J., Gordon S., 1960, Technical Report, An Analytical Investigation of Three General Methods of Calculating Chemical Equilibrium Compositions
Zeleznik F. J., Gordon S., 1968, Ind. Eng. Chem., 60, 27
gnu.org, 2007, GNU General Public License, https://www.gnu.org/licenses/gpl-3.0.en.html
mcs.anl.gov, 1997, The Message Passing Interface (MPI) standard, www.mcs.anl.gov/research/projects/mpi/index.htm
openmp.org, 1997, The OpenMP API specification for parallel programming, http://www.openmp.org/
van Zeggeren F., Storey S. H., 1970, The Computation of Chemical Equilibria, Cambridge University Press, Cambridge, UK
van't Hoff J., 1884, Études de Dynamique chimique (in French), Amsterdam, Frederik Muller & Co.

APPENDIX A: EVALUATION OF PRODUCTS AND SCALING

Throughout our calculations, expressions of the form

$$p_{ij} = K_i \prod_{\substack{l\in\mathcal{E}_0\\ l\neq j}} n_l^{\nu_{il}} \tag{A1}$$

are ubiquitous in the algorithm. The factors in equation (A1) can differ in hundreds of orders of magnitude. To evaluate these products and avoid numerical overflow, we use the equivalent expression

$$p_{ij} = \exp\left\{ \ln K_i + \sum_{\substack{l\in\mathcal{E}_0\\ l\neq j}} \nu_{il} \ln n_l \right\} . \tag{A2}$$

Additionally, we define

$$\psi_j := \max_{i\in\mathcal{S}\setminus\mathcal{E}} \left(\ln K_i + \sum_{\substack{l\in\mathcal{E}_0\\ l\neq j}} \nu_{il} \ln n_l \right) - \xi_j \tag{A3}$$

where $\xi_j \geqslant 0$ are constants. Since the solution of the equation

$$P_j(n_j) = 0 \tag{A4}$$

is invariant against multiplication with a scaling factor e^{ψ_j}, we can write

$$\hat{P}_j(n_j) = \sum_{k=0}^{N_j} \hat{A}_{jk} n_j^k = e^{\psi_j} P_j(n_j) = 0 \tag{A5}$$

with the coefficients

$$\hat{A}_{j0} = e^{-\psi_j} A_{j0} = e^{\ln n_{j,\min} - \psi_j} - e^{\ln(\epsilon_j n_{(\mathrm{H})}) - \psi_j} \tag{A6}$$

$$\hat{A}_{j1} = e^{-\psi_j} A_{j1} = e^{-\psi_j} + \sum_{\substack{i\in\mathcal{S}\setminus\mathcal{E}\\ \nu_{ij}=1\\ \epsilon_i=\epsilon_j}} \exp\left\{ \ln K_i + \sum_{\substack{l\in\mathcal{E}\\ l\neq j}} \nu_{il} \ln n_l - \psi_j \right\} , \tag{A7}$$

$$\hat{A}_{jk} = e^{-\psi_j} A_{jk} = k \sum_{\substack{i\in\mathcal{S}\setminus\mathcal{E}\\ \nu_{ij}=k\\ \epsilon_i=\epsilon_j}} \exp\left\{ \ln K_i + \sum_{\substack{l\in\mathcal{E}\\ l\neq j}} \nu_{il} \ln n_l - \psi_j \right\} . \tag{A8}$$

APPENDIX B: ANALYTIC EXPRESSION OF THE ELECTRON DENSITY

Consider the case $|\nu_{i0}| \leqslant 1$. Then, in analogy to equation (34), we write

$$0 = n_0 \left(1 + \sum_{\substack{i\in\mathcal{S}\setminus\mathcal{E}\\ \nu_{i0}=-1}} K_i \prod_{j\in\mathcal{E}} n_j^{\nu_{ij}} \right) - \frac{1}{n_0} \sum_{\substack{i\in\mathcal{S}\setminus\mathcal{E}\\ \nu_{i0}=1}} K_i \prod_{j\in\mathcal{E}} n_j^{\nu_{ij}} \tag{B1}$$

This equation can be solved for n_0 directly and it follows

$$n_0 = \sqrt{\frac{\alpha}{\beta}} . \tag{B2}$$

with

$$\alpha = \sum_{\substack{i\in\mathcal{S}\setminus\mathcal{E}\\ \nu_{i0}=1}} K_i \prod_{j\in\mathcal{E}} n_j^{\nu_{ij}} \tag{B3}$$

and

$$\beta = 1 + \sum_{\substack{i\in\mathcal{S}\setminus\mathcal{E}\\ \nu_{i0}=-1}} K_i \prod_{j\in\mathcal{E}} n_j^{\nu_{ij}} . \tag{B4}$$

This paper has been typeset from a TEX/LATEX file prepared by the author.